WHAT IS CLAIMED IS:

1. A method for determining whether documents, in a large collection of documents, are near-duplicates, the method comprising:

a) for each of at least some of the documents in the large collection of documents, generating at least two fingerprints;

b) preprocessing the fingerprints to identify any fingerprints that are associated with only one document; and

c) determining whether or not documents are near-duplicate documents based on fingerprints other than those identified as being associated with only one document.

2. The method of claim 1 wherein the act of determining whether or not documents are near-duplicate documents includes:

i) for any two documents, determining whether or not any fingerprints of a first of the two documents matches any fingerprints of a second of the two documents, and

ii) if it is determined that a fingerprint of the first of the two documents does match a fingerprint of the second of the two documents, then concluding that the two documents are near-duplicates.

3. The method of claim 1 wherein the act of generating at least two fingerprints for each of the documents includes:

i) extracting parts from the document,

ii) hashing each of the extracted parts to generate a hash value for each of the extracted parts,

iii) populating a predetermined number of lists with the extracted parts based on their respective hash values, and

iv) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list.

4. The method of claim 3 wherein the act of hashing each of the extracted parts to generate a hash value for each of the extracted parts uses a hash function that is repeatable, deterministic and not sensitive to state.

5. The method of claim 3 wherein the parts extracted from the document are selected from a group of parts consisting of characters, words, sentences, paragraphs and sections.

6. The method of claim 3 wherein the parts extracted from the document do not overlap.

7. The method of claim 3 wherein the parts extracted from the document overlap.

8. The method of claim 3 wherein each of the acts of determining a fingerprint uses a hashing function with a low probability of collision.

9. The method of claim 3 wherein the act of determining a fingerprint uses a function that is sensitive to an order of the parts within a list.

10. The method of claim 3 wherein the act of determining a fingerprint uses a function that is insensitive to an order of the parts within a list.

11. An apparatus for determining whether documents, in a large collection of documents, are near-duplicates, the apparatus comprising:

a) a fingerprint generator for generating, for each of the documents in the large collection of documents, at least two fingerprints;

b) a preprocessor for identifying any fingerprints that are associated with only one document; and

c) a fingerprint comparison facility for determining whether or not documents are near-duplicate documents based on fingerprints other than those identified as being associated with only one document.

12. The apparatus of claim 11 wherein the fingerprint generator includes:

i) an extractor for extracting parts from the document,

ii) a hashing facility for hashing each of the extracted parts to generate a hash value for each of the extracted parts,

iii) list population facility for populating a predetermined number of lists with the extracted parts based on their respective hash values, and

iv) means for determining a fingerprint for each of the predetermined number of lists, based on the contents of the list.

13. A method for clustering documents, the method comprising:

a) for each of the documents, generating at least two fingerprints; and

b) for each of the documents,

i) determining whether or not the document is a near-duplicate of any of previously processed documents, based on fingerprints of the documents,

ii) if it is determined that the document is not a near-duplicate of any previously processed document, then associating the document with a unique cluster identifier, and

iii) if it is determined that the document is a near-duplicate of a previously processed document, then associating the document with a cluster identifier associated with the previously processed document.

14. A method for filtering search results to remove near-duplicates, the method comprising:

a) for each of a predetermined number of candidate search results, determining whether the candidate search result is a near-duplicate of another candidate search result; and

b) if it is determined that the candidate search result is a near-duplicate of another candidate search result, then rejecting the candidate search result.

15. The method of claim 14 wherein the act of determining whether a candidate search result is a near-duplicate of another candidate search result includes

i) comparing a cluster identifier of the candidate search result with that of the other candidate search result, and

ii) if the cluster identifiers of the two candidate search results match, then concluding that the two candidate search results are near-duplicates.

16. The method of claim 15 wherein cluster identifiers of the candidate search results are assigned by:

i) determining whether or not a document corresponding to the candidate search result is a near-duplicate of any of previously processed documents,

ii) if it is determined that the document corresponding to the candidate search result is not a near-duplicate of any previously processed document, then associating the document with a unique cluster identifier, and

iii) if it is determined that the document corresponding to the candidate search result is a near-duplicate of a previously processed document, then associating the document corresponding to the candidate search result with a cluster identifier associated with the previously processed document.

17. A method for determining whether two documents are near-duplicates, the method comprising:

a) for each of the two documents, generating at least two fingerprints by

i) extracting parts from the document,

ii) hashing each of the extracted parts to generate a hash value for each of the extracted parts,

iii) populating at least two lists with the extracted parts based on their respective hash values, and

iv) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list; and

b) determining whether or not the two documents are near-duplicate documents based on their fingerprints.

18. The method of claim 17 wherein the act of determining whether or not the two documents are near-duplicate documents includes:

i) determining whether or not any fingerprints of a first of the two documents matches any fingerprints of a second of the two documents, and

ii) if it is determined that a fingerprint of the first of the two documents does match a fingerprint of the second of the two documents, then concluding that the two documents are near-duplicates.

19. The method of claim 17 wherein the act of hashing each of the extracted parts to generate a hash value for each of the extracted parts uses a hash function that is repeatable, deterministic and not sensitive to state.

20. The method of claim 17 wherein the parts extracted from the document are selected from a group of parts

consisting of characters, words, sentences, paragraphs and sections.

21. The method of claim 17 wherein the parts extracted from the document do not overlap.

22. The method of claim 17 wherein the parts extracted from the document overlap.

23. The method of claim 17 wherein the act of determining a fingerprint uses a hashing function with a low probability of collision.

24. The method of claim 17 wherein the act of determining a fingerprint uses a function that is sensitive to an order of the parts within a list.

25. The method of claim 17 wherein the act of determining a fingerprint uses a function that is insensitive to an order of the parts within a list.

26. A method, for use in a crawling facility, for reducing processing and bandwidth used, the method comprising:

a) for each of the documents, generating at least two fingerprints by

i) extracting parts from the document,

ii) hashing each of the extracted parts to generate a hash value for each of the extracted parts,

iii) populating at least two lists with the extracted parts based on their respective hash values, and

iv) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list;

b) determining whether or not the two documents are near-duplicate documents based on their fingerprints; and

c) if it is determined that the two documents are near-duplicates, then indicating that one of the two documents is not to be processed during a subsequent crawl.

27. A method for treating broken links to document, the method comprising:

a) determining whether a link to a first document is broken;

b) if it is determined that a link to a first document is broken, determining whether there exists a second document that is a near-duplicate of the first document; and

c) if it is determined that there exists a second document that is a near-duplicate of the first document, then replacing the broken link to the first document with a link to the second document,

wherein the act of determining whether or not there exists a second document is a near-duplicate of the first document is performed by:

i) for each of the documents, generating at least two fingerprints by

A) extracting parts from the document,

B) hashing each of the extracted parts to generate a hash value for each of the extracted parts,

C) populating at least two lists with the extracted parts based on their respective hash values, and

D) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list; and

ii) determining whether or not the two documents are near-duplicate documents based on their fingerprints.

28. An apparatus for determining whether two documents are near-duplicates, the apparatus comprising:

a) a fingerprint generator for generating at least two fingerprints for each of the two documents, the fingerprint generator including

i) an extractor for extracting parts from the document,

ii) a hashing facility for hashing each of the extracted parts to generate a hash value for each of the extracted parts,

iii) a list population facility for populating at least two lists with the extracted parts based on their respective hash values, and

iv) means for determining, for each of the predetermined number of lists, a fingerprint based on the contents of the list; and

b) a comparison facility for determining whether or not the two documents are near-duplicate documents based on their fingerprints.

29. An improved crawling facility, for reducing processing and bandwidth used, the crawling facility comprising:

a) a fingerprint generator for generating, for each of the documents, at least two fingerprints, the fingerprint generator including

i) an extractor for extracting parts from the document,

ii) a hashing facility for hashing each of the extracted parts to generate a hash value for each of the extracted parts,

iii) a list population facility for populating at least two lists with the extracted parts based on their respective hash values, and

iv) means for determining, for each of the predetermined number of lists, a fingerprint based on the contents of the list;

b) a comparison facility for determining whether or not the two documents are near-duplicate documents based on their fingerprints; and

c) a document processor, wherein if it is determined that the two documents are near-duplicates, then the document processor indicates that one of the two documents is not to be processed during a subsequent crawl.

30. A search filter for processing search results to remove near-duplicates, the search filter comprising:

a) a near-duplicate determination facility for determining, for each of a predetermined number of candidate search results, whether the candidate search result is a near-duplicate of another candidate search result; and

b) a filter for rejecting the candidate search result if it is determined that the candidate search result is a near-duplicate of another candidate search result.

31. The search filter of claim 30 wherein the near-duplicate determination facility includes a comparison facility for comparing a cluster identifier of the candidate search result with that of another candidate search result, and wherein if the cluster identifiers of the two candidate search results match, then it is concluded that the two candidate search results are near-duplicates.

32. A machine-readable medium having stored thereon a plurality of records, each of the records comprising:

a) a first field for storing a document identifier; and

b) a plurality of lists, each of the plurality of lists containing elements of a document identified by the document identifier stored in the first field,

wherein a hash function is used to determine which of the plurality of lists each of the elements will be contained in.

33. A machine-readable medium having stored thereon a plurality of records, each of the records comprising:

a) a first field for storing a document identifier; and

b) a plurality of fingerprints, wherein each of the fingerprints is a low collision probability hash function of elements contained in a corresponding list, and wherein the elements are elements of a document identified by the document identifier stored in the first field.

34. A machine-readable medium having stored thereon machine-executable instructions which, when executed by a machine:

a) extract parts from a document,

ii) hash each of the extracted parts to generate a hash value for each of the extracted parts,

iii) populate a predetermined number of lists with the extracted parts based on their respective hash values, and

iv) for each of the predetermined number of lists, determine a fingerprint based on the contents of the list.

35. A method for generating at least two fingerprints for a document comprising:

a) extracting parts from the document;

b) hashing each of the extracted parts to generate a hash value for each of the extracted parts;

c) populating a predetermined number of lists with the extracted parts based on their respective hash values; and

d) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list.

36. The method of claim 35 wherein each of the lists has an associated hashing function,

wherein each of the extracted parts can be contained in none of the lists, one of the lists, or more of the lists based on the hash functions for the lists.

37. The method of claim 36 wherein for each hash function is dynamically adjusted such that the probability that the hash function will populate its associated list with a part decreases as the size of the document increases.

38. A method comprising:

a) determining whether there exists a second document that is a near-duplicate of a first document; and

b) indexing the first document but not the second document,

wherein the act of determining whether or not there exists a second document is a near-duplicate of the first document is performed by:

i) for each of the documents, generating at least two fingerprints by

A) extracting parts from the document,

B) hashing each of the extracted parts to generate a hash value for each of the extracted parts,

C) populating at least two lists with the extracted parts based on their respective hash values, and

D) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list; and

ii) determining whether or not the two documents are near-duplicate documents based on their fingerprints.

39. A method for determining whether two documents are near-duplicates, the method comprising:

a) for each of the two documents, generating at least two fingerprints; and

b) determining whether or not the two documents are near-duplicate documents by

i) determining whether or not any one of the fingerprints of a first of the two documents matches any one of the fingerprints of a second of the two documents, and

ii) if it is determined that any one fingerprint of the first of the two documents does match any one fingerprint of the second of the two documents, then concluding that the two documents are near-duplicates.

40. A method for determining whether two objects are near-duplicates, the method comprising:

a) for each of the two objects, generating at least two fingerprints by

i) extracting features from the object,

ii) hashing each of the extracted features to generate a hash value for each of the extracted features,

iii) populating at least two lists with the extracted features based on their respective hash values, and

iv) for each of the predetermined number of lists, determining a fingerprint based on the contents of the list; and

b) determining whether or not the two objects are near-duplicates based on their fingerprints.

41. The method of claim 40 wherein each of the two objects is a word, and

wherein the extracted features define context vectors.

42. The method of claim 40 wherein each of the two objects is a word, and

wherein, in each case, the extracted features are words that frequently occur in close proximity to the word.

43. The method of claim 40 wherein the two objects are words, and

wherein if the two objects are determined to be near duplicates, then determining the two words to be synonyms.

44. A method for determining whether a first document and a second document in a collection of documents are near-duplicates, the method comprising:

a) for each of the documents in the collection of documents, generating at least two fingerprints; and

b) concluding that the first and second documents are near-duplicates if any one of the at least two fingerprints of the first document matches any one of the at least two fingerprints of the second document,

wherein documents in the collection of documents without any common fingerprints are not checked to determine whether or not they are near duplicates.

45. The method of claim 44 further comprising:

a2) for each of the documents in the collection of documents, generating a document-fingerprint pair for each of the at least two fingerprints; and

a3) sorting the fingerprint-document pairs based on values of the fingerprints.